Saskatchewan
Research Council

GRAPHICS SOFTWARE FOR DATA ANALYSIS:

I. SYSTEM DESIGN

by

M. Mellinger
and
J.W.J. Smith

SRC Technical Report No. 163

Publication R-851-3-B-84

October, 1984

GRAPHICS SOFTWARE FOR DATA ANALYSIS:

I. SYSTEM DESIGN

by

M. Mellinger
and
J.W.J. Smith

SRC Technical Report No. 163

R-851-3-B-84

October, 1984

GRAPHICS SOFTWARE FOR DATA ANALYSIS

I. SYSTEM DESIGN

by

M. Mellinger
and
J.W.J. Smith

The work reported here was carried out as part of the Geochemistry and Data Analysis project at the Saskatchewan Research Council.

This report is one from the series GRAPHICS SOFTWARE FOR DATA ANALYSIS. Information on other reports from this series may be obtained by inquiring to the:

Saskatchewan Research Council
30 Campus Drive
Saskatoon, Saskatchewan
S7N 0X1

October, 1984

SRC Publication No. R-851-3-B-84

TABLE OF CONTENTS

		Page
I.	INTRODUCTION	1
II.	DATA FILE DESIGN	3
	1. The Lithogeochemical Data File	3
	2. The Binary Data File	4
	3. The Compressed LDF	7
III.	COMPUTER ENVIRONMENT DESIGN	10
	1. Components	10
	2. Integration	12
IV.	GRAPHICS SOFTWARE DESIGN	15
	1. General Design Elements	15
	2. Software Components	16
V.	CONCLUDING REMARKS	18
	REFERENCES	20

I. INTRODUCTION

The purpose of the system we developed is to provide geochemists with effective means of investigating their data, or in other terms of extracting information from their data.

The fact that geochemists are considered here is more due to the fact that geochemistry is our main field of application, than to particular problems geochemists may face during their data analysis work. The various issues discussed in this report actually apply to any data analysis situation.

Considering design as a process which defines critical issues and identifies responses to these (see Peters, 1981, Section 2.4), our system must respond to three main critical issues:

(1) analytical data and their complementary information, here the "geochemical data", must be organized in a manner suitable for data analysis;

(2) investigation of the geochemical data, and the data analysis procedure, must be possible both at a multivariate level and at a graphical level; and

(3) the data analysis procedure must be easy to use by people inexperienced with computers.

The responses to these three critical issues, presented in detail in this report, are, respectively:

(1) the design of the Lithogeochemistry Data File and its binary equivalent;

(2) the coordinated usage of a mainframe computer and a desk-top graphics microcomputer; and

(3) the design of interactive software which leaves the user with maximum flexibility of operation (to date, this has been limited to the desk-top environment).

The first two of these responses were summarized in an earlier presentation (Mellinger, 1982). The third response and some comments about the possible effects of future developments of desk-top computers are also included in this report.

II. DATA FILE DESIGN

The purpose of the data file is to organize geochemical data in a manner suitable for their analysis. The Lithogeochemical Data File (LDF), an ASCII file format, was designed to achieve this purpose. A complete presentation of the LDF is given in Mellinger (1983).

II.1 The Lithogeochemical Data File

Critical issues for the LDF were identified as:

(1) exhaustiveness: both analytical data and complementary information relevant to the geochemist's problems should be included;

(2) compatibility with the Saskatchewan Energy and Mines Geochemical Data File (Dunn, 1978);

(3) compactness: for easy manipulation and usage with data analysis software; and

(4) flexibility: to answer unforseen and future needs.

The first two critical issues determine the minimum amount of complementary information that must be coded in the LDF.

The third critical issue, compactness, was answered by using a record-oriented file format, a dual-file structure, and a compact coding scheme (Figure 1). The fixed format of the various records that describe each sample, avoids the presence of field qualifiers in the file. The dual-file structure is a practical way of responding to the following, somewhat contradictory, functions of a data file: to be exhaustive enough

for proper archiving, and compact enough for easy manipulation and investigation. Information that is not directly relevant to data analysis is stored in the Header_file, whereas analytical data and complementary information relevant to data analysis are stored in the Master_file. Finally, the compact_coding_scheme makes use of integer hierarchical (low-redundancy) coding for complementary information, and of compact coding (four ASCII characters per element) for the analytical data.

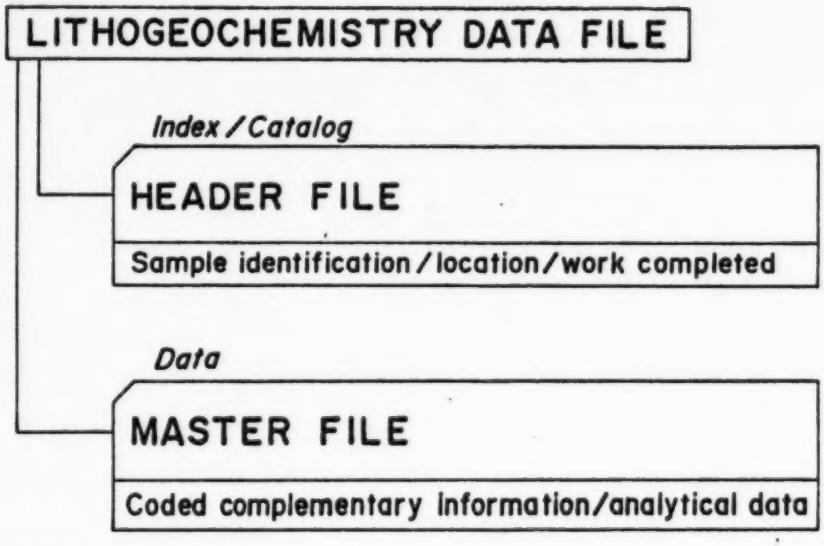
The last critical issue, flexibility, was answered both internally, by allowing some_user-defined_codes and keeping some_fields open for custom-coding, and in terms of expansion, by allowing_new_record-types to be defined.

Each 80-column record is uniquely identified by a record-type code (cols. 79-80) and by a sample number (cols. 1-8). Correspondence between records for a given sample is assured by the sample number, both within the Master file and between Header file and Master file. Record-type codes defined to date are 99 (Header file), and 01 to 06 (Master file).

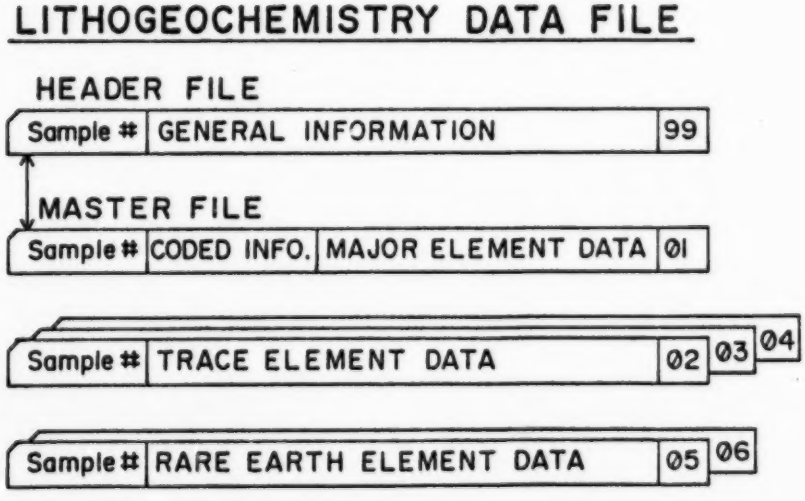
The usefulness of the LDF has been optimized for data analysis. Nevertheless, the LDF also provides a valid means of storing all information necessary for the management and archiving of geochemical data.

II.2 The Binary Data File

The need for a binary data file format arose when data processing in a desk-top environment was initiated. Data stored in binary format occupy less storage space, and do not require conversion before processing, thus saving processing time as well. Both savings become significant when a desk-top (micro) computer is used.



(a) Purpose of the dual-file structure.


LITHOGEOCHEMISTRY DATA FILE
HEADER FILE
Sample # GENERAL INFORMATION 99
MASTER FILE
Sample # CODED INFO. MAJOR ELEMENT DATA 01
Sample # TRACE ELEMENT DATA 02 03 04
Sample # RARE EARTH ELEMENT DATA 05 06

(b) Records included in the LDF.

Figure 1. General structure of the LDF.

The critical issue in this case was that all information about the file be self-contained. The required file information consists of:

(1) the file identification;

(2) general size parameters;

(3) information about the variable types present; and

(4) identification of the variables.

The design elements of each of the above items are:

(1) a file_title (alphanumeric string);

(2,3) the following file_parameters: number of samples (rows), number of variables (columns), number of alphanumeric variables, number of discrete variables. The number of continuous variables can be deduced from these parameters (= total - alphanumeric - discrete);

(4) a variable_name dictionary (alphanumeric string).

The various file parameters are more easily understood if the data file is viewed as a data table (Figure 2).

As no individual variable qualifiers are used, the variables are assumed to be ordered as follows: alphanumeric variables first, then discrete variables, then continuous variables. In our case, the alphanumeric variables include selected sub-fields of the sample number (e.g. project mnemonics), and rock-type or sample-type mnemonics. The discrete variables include the complementary information coded as integer values. The continuous variables are measured variables, the analytical variables in our case. Constraints on data processing for these variable types are summarized in Table 1.

The variable name dictionary consists of an alphanumeric string in which the names of all variables (four characters each) are concatenated in the proper order.

Thus, upon accessing a binary data file in the designed format, all programs read first the file title and file parameters, then decode the dictionary string according to the number of variables, and finally read the sample records according to the file parameters. These file parameters are also used during data processing when specific program options are reserved for specific variable types.

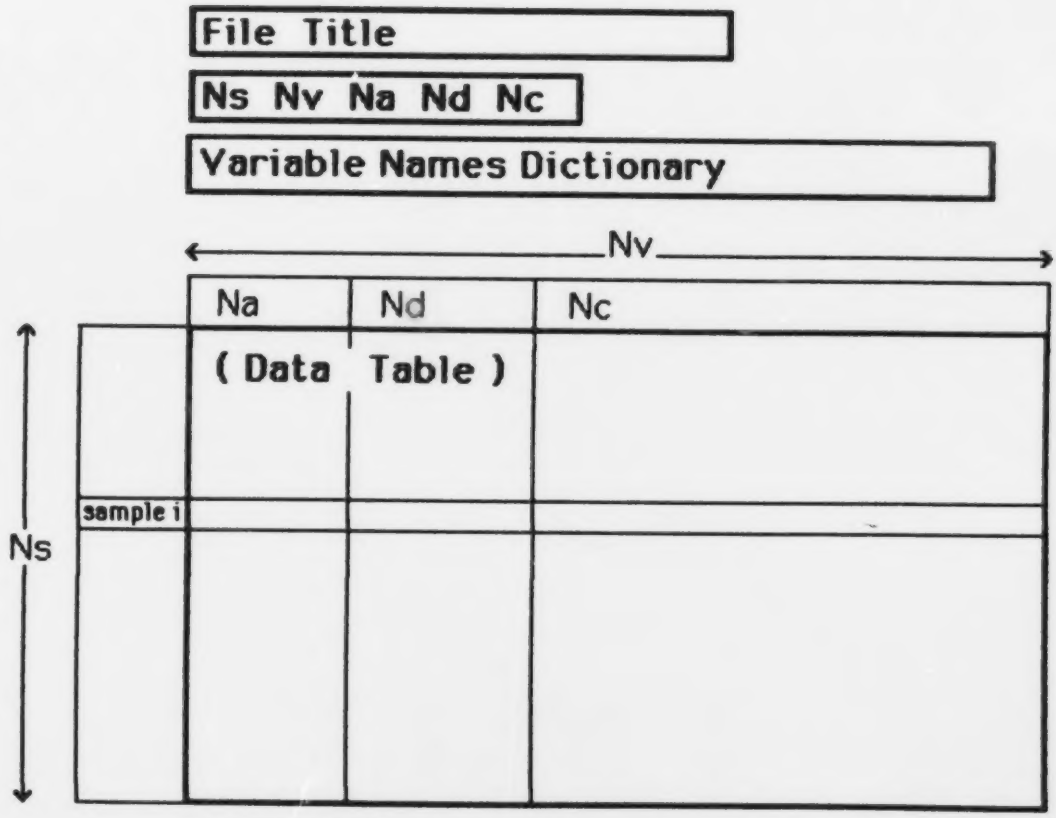
II.3 The Compressed LDF

The transition from an LDF to the equivalent binary data file is accomplished with a compressed LDF.

The compressed LDF is essentially the ASCII form of the binary data file, in which each sample record is a continuous string devoid of blank fields. However, additional parameters are required in the compressed LDF. These are added to the other file parameters and are:

(1) the number of major elements analyzed for, which is required for the translation procedure. Major elements are coded in F4.2 implicit format, whereas trace elements are coded in F4.0 format; the number of trace elements can be deduced from this and other parameters by the translation program;

(2) the record length of the sample records, which is required as a data transfer parameter.



Ns = number of samples (rows)
Nv = number of variables (columns)
Na = number of alphanumeric variables
Nd = number of discrete variables
Nc = number of continuous variables

Figure 2. Content of a binary data file (in bold) in terms of the structure of the data table.

Table 1. Constraints on data processing for the various variable types used.

Variable Type	Usage in data processing				
	Univariate Statistics	Subfile Selection[1]	Contingency Tables	Histograms	Bivariate Plots
Alphanumeric	partially listed only	-	-	-	as illustrative variable only
Discrete	yes	yes	yes	yes	yes
Continuous	yes	yes	-	yes	yes

[1] in terms of using the variable to define a selection criterion.

III. COMPUTER ENVIRONMENT DESIGN

Generally, computer systems must be designed to suit particular applications under given operational constraints. In our case, the critical issues are related to the capability for:

(1) effective multivariate data analysis, making extensive use of techniques such as factor analysis and hierarchical classification methods; and

(2) easy-to-use interactive graphical data analysis both at the level of the data and at the level of the results from multivariate data analysis.

III.1 Components

The first function, multivariate data analysis, must be implemented on a mainframe computer, because the related software is available for this environment only. The second function, interactive graphical data analysis, may also be implemented on a mainframe computer; our operational constraints, however, led us to the use of a desk-top graphics microcomputer. Table 2 summarizes the characteristics of both types of computers and indicates critical characteristics in our case.

The computer environment selected consists of a combination of a mainframe computer and a graphics microcomputer, with classes of tasks attributed to each one. Tasks performed using the mainframe component are those which require large memory space for direct use or for access during data processing; such tasks are: extensive data management, and heavy data processing due to either large amounts of data or fairly involved calculations. Tasks reserved for the graphics microcomputer are those which are best accomplished in an interactive mode, and are essentially graphical in nature.

Table 2. Comparison of characteristics of a mainframe computer and a graphics microcomputer (from Mellinger, 1982).

Mainframe Computer	Graphics Microcomputer
Hardware:	
(*) large memory space (eventually virtual)	- limited memory space
(*) large data storage available	- large data storage possible
(*) high processing speed	- low processing speed
System Design:	
- multi-purpose processing and outputs	(*) specialized graphic outputs
Software:	
- mostly available on the market	- mostly developed by user
Status of User:	
- customer of the computer center	- owner/renter
Cost to User:	
- no direct acquisition cost (hardware)	- acquisition cost (hardware)
- software cost variable	- high software cost (mainly in man-years)
- high operating cost, rates based on system usage (CPU, connect, storage, I/O, ...)	(*) low operating cost

Notes: (*) indicates which characteristics of each computer environment were critical in the design of the computer system shown in Figure 3.

CPU = central processing unit (core memory), I/O = input/output operations and devices.

Figure 3 shows how a typical lithogeochemical exploration project can be carried out in the computer environment designed here.

III.2 Integration

In order to integrate both mainframe computer and microcomputer into one system, electronic communication channels and procedures, and a common file format must be defined. These include (Figure 4) a direct-line link between the two computers, the compressed LDF format as standard ASCII file format, a standardized procedure for the transfer of a compressed LDF, and a standardized data translation procedure at the microcomputer level. Note that data transfer is accomplished from the mainframe computer to the microcomputer only, as long-term data storage and multivariate data analysis are implemented on the former. Some data management capability is included in the microcomputer component, which consists in flexible subfile selection using either row or column criteria, or both. This aspect of data management is seen as an integral part of the data analysis procedure, whereby sub-data tables of required homogeneity are generated as the investigation proceeds to higher resolution levels.

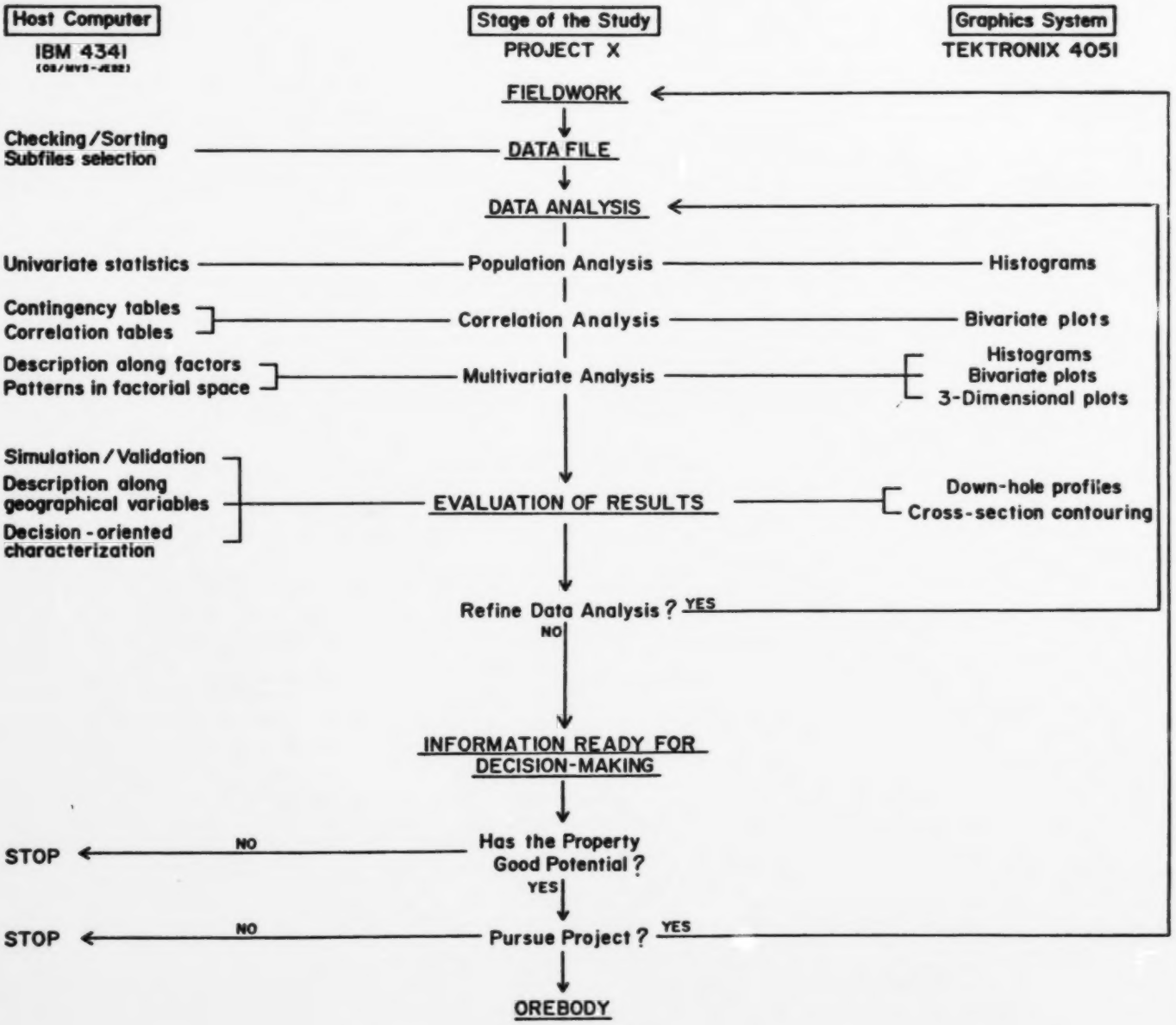


Figure 3. A computer environment adapted to lithogeochemical investigations, applied to mineral exploration (from Mellinger, 1982).

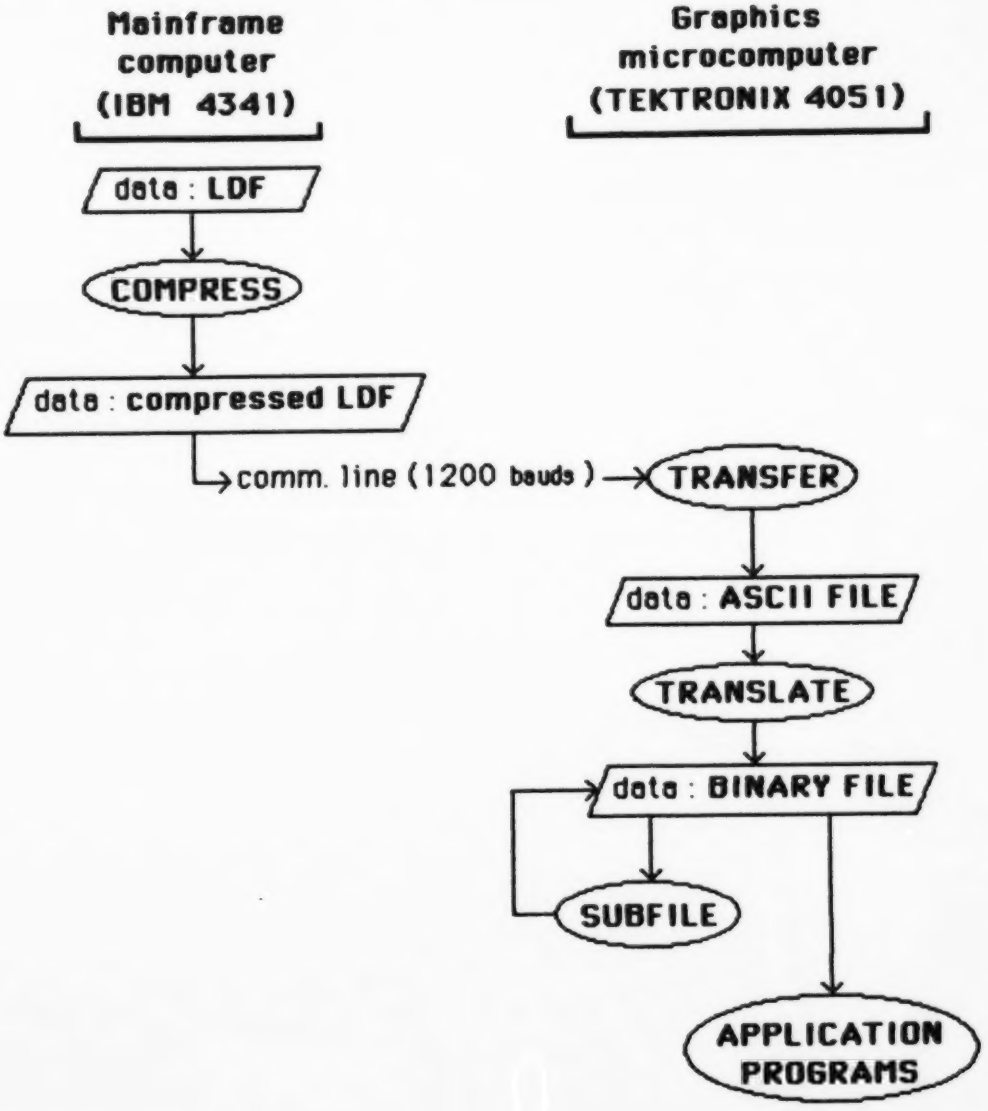


Figure 4. Integration of the two components of the designed computer system.

IV. GRAPHICS SOFTWARE DESIGN

Interactive graphical data analysis is implemented on the graphics microcomputer. The critical issues are related to the typical user who:

(1) is inexperienced with computers;

(2) expects to obtain graphics, and to modify their parameters easily;

(3) wishes to produce a hard copy of graphics once it is found informative, either for reference or for inclusion into a report.

IV.1. General Design Elements

Computer inexperience is answered by designing software which guides the user along and is menu driven at all levels of its operation. "Foolproofing" of the software with respect to unsuitable input from the user must also be included.

The level of expectation of the user with respect to ease of use of the software is met by ensuring, firstly, that the user requires no knowledge of the file structure and internal operation of the software, and, secondly, that software operation answers the user's needs. For example, all the user must know about the data file is its location on the storage medium, i.e., its file number; the data file itself contains all the information necessary for its use (see section II.2). Also, for example in the case of the histogram program, all the user must do before obtaining the first histogram, is select the variable for which the histogram is desired, as prompted by the program; after this, the user can modify the histogram parameters at will through the various menus.

The last expectation of the user, the possibility of obtaining hard copies, is met by including flexible hard copy output options in the software, For example, in the case of the histogram program, the user may obtain either a hard copy of the screen output for reference purposes (containing the histogram with detailed statistics and specifications), or a good-quality plot of up to seven histograms on a standard (8½ x 11 inches) sheet with some statistics and annotations, for reporting purposes.

IV.2. Software Components

The data file under investigation is either the initial file transferred from the mainframe computer or a subfile created locally using the SUBFILE program, as discussed in section III.2.

The programs developed on the graphics microcomputer to date are as follows:

(1) SCANSTAT: produces a printed listing of the file characteristics, of univariate descriptive statistics for all numeric variables, and a partial list of the content of alphanumeric variables. This information gives the user a general idea about the file content.

(2) CONTINGENCY: produces contingency tables (two-way tables) for any pair of discrete variables selected by the user.

(3) HISTOGRAM: produces a histogram for any numeric variable selected by the user. Histogram parameters that can be adjusted are: minimum value, maximum value, cell increment, frequency scale, file label, histogram label, and sub-population selection criteria. Detailed statistics are displayed for each cell and for the total

histogram on the screen. Plotter output is available with up to seven histograms plotted at user-selected locations on a standard (8½ x 11 inches) sheet.

(4) BIVARIATE: produces a bivariate plot for any pair of numeric variables selected by the user. Plot parameters that can be adjusted are: minimum and maximum values on either axis, file label, axes labels, plotted symbols, and sub-population selection criteria. Options for plotted symbols include: single symbol selection, plotting of the values of any alphanumeric or discrete variable as symbols, and plotting of user-selected symbols for user-defined ranges of values for any discrete or continuous variable. Plotter output is available, and the user has a choice of plot shape, size, and location on a standard (8½ x 11 inches) sheet.

Details about the design and operation of these programs are given in separate reports (Smith and Mellinger, 1983a, b; 1984).

V. CONCLUDING REMARKS

The three main critical issues addressed by our system are:

(1) data organization suitable to data analysis;

(2) effective and flexible data analysis at all levels;

(3) ease of use of the system.

We believe that we responded to these critical issues as demonstrated, for example, by the practical comprehensiveness and flexibility shown by the data files designed, by the increased effectiveness and efficiency of geochemical data analysis with the system developed, and by the fact that the graphics programs were used successfully within a short period of time by users who had no prior experience with computers.

Desk-top computers are evolving rapidly. Their data processing power is increasing to a point where many tasks performed only on mainframe computers until now will soon become practical on desk-top computers. The graphics micro computer we use now has 32K of core memory and is operated by an 8-bit processor; new generation desk-top computers now work with 512K core memory and are powered by 32/16-bit processors, with true 32-bit processors not far off in the future. Also, decreasing screen (CRT) costs now permit inclusion of high-resolution graphics screens (e.g., 80 pixels per inch) in affordable desk-top computers. Finally, with the increasing number of customers who are not familiar with the internal workings of computers (nor wish to become familiar with these!), computer manufacturers and software developers are beginning to offer systems with truly user-oriented interfaces which let users do their work <u>with</u> the help of the computer, <u>not despite</u> the computer and its operating system.

How will these developments affect our system design? If one tries to answer this question in terms of the system's three critical issues, the following come to mind:

(1) data organization may or may not evolve, although the potential implementation of powerful data base management systems on desk-top computers may offer interesting flexibility and handling power;

(2) multivariate data analysis may soon become feasible under reasonable conditions on a desk-top computer, and thus all functions of of our system may become integrated on one desk-top computer without the use of a mainframe computer; however, software development in the desk-top environment for multivariate techniques may take significant effort if re-writing of the software is involved;

(3) finally, the user will most likely gain much as a result of all these developments, provided system design makes ease of use a high priority.

REFERENCES

Dunn, C.E. (1978). Geochemical data file of Northern Saskatchewan. In: Summary of Investigations 1978, J.E. Christopher and R. MacDonald eds., Sask. Geol. Survey Miscell. Report 78-10, p. 141-144.

Mellinger, M. (1982). New developments in SRC's lithogeochemistry program. In: Summary of Investigations 1982, J.E. Christopher and R. MacDonald (eds.), Saskatchewan Energy and Mines Miscell. Report 82-4, p.60-63.

————(1983). The Lithogeochemistry Data File: user manual. SRC Publication No. R-740-1-E-83, 46 pages.

Peters, L.J. (1981). Software Design: Methods and Techniques. Yourdon Press (New York), 234 pages.

Smith, J.W.J., Mellinger, M. (1983a). Graphics software for data analysis: III. Histogram program-user's guide. SRC Technical Report #146 (also SRC Publication R-740-5-B-83), 33 pages.

————(1983b).————: IV Bivariate plot program-user's guide. SRC Technical Report #147 (also SRC Publication R-740-9-B-83), 45 pages.

————(1984).————: II Data transfer and data file management-user's guide. SRC Technical Report #156 (also SRC Publication R-851-2-B-84), 41 pages.